

another variant of memikatz.exe called netkatz.exe.

This tool will go online and fetch memikatz.exe and then run it in memory:

Since it goes to the internet, lets check our configuration if its logging any network connection:

```
Current configuration:
- Service name: Sysmon
- Driver name: SysmonDrv
- HashingAlgorithms: SHA1,MD5,SHA256,IMPHASH
- Network connection: disabled
- Image loading: disabled
- CRL checking: disabled
- Process Access: enabled

Rule configuration (version 3.40):
- ProcessCreate          onmatch: include
- FileCreateTime         onmatch: include
- NetworkConnect         onmatch: include
- ProcessTerminate       onmatch: include
- DriverLoad             onmatch: include
- ImageLoad              onmatch: include
- CreateRemoteThread     onmatch: include
  TargetImage            filter: image
                        value: 'lsass.exe'
- RawAccessRead          onmatch: include
- ProcessAccess          onmatch: include
  TargetImage            filter: image
                        value: 'lsass.exe'
- FileCreate             onmatch: include
- RegistryEvent          onmatch: include
- RegistryEvent          onmatch: include
- RegistryEvent          onmatch: include
- FileCreateStreamHash   onmatch: include
- PipeEvent              onmatch: include
- PipeEvent              onmatch: include
```

Here we see the Network connection and Image loading is disabled, Image loading is when a file is loaded into the memory.

Lets edit our configuration file to log Network connection and image loading:

In the network connection we change the onmatch="include" to onmatch="exclude" which we tell the Sysmon to exclude nothing and log everything as we want to log all the network connection, and we do the same for ImageLoad events too so any file that is loaded into the memory is logged:

```
<File Edit Format View Help>
<Sysmon schemaversion="3.40">
<!-- Capture all hashes -->
<hashAlgorithms></hashAlgorithms>
<EventFiltering>
<!-- Event ID 1 == Process Creation. -->
<ProcessCreate onmatch="include"/>
<!-- Event ID 2 == File Creation Time. -->
<FileCreateTime onmatch="include"/>
<!-- Event ID 3 == Network Connection. -->
<NetworkConnect onmatch="exclude"/>
<!-- Event ID 5 == Process Terminated. -->
<ProcessTerminate onmatch="include"/>
<!-- Event ID 6 == Driver Loaded. -->
<DriverLoad onmatch="include"/>
<!-- Event ID 7 == Image Loaded. -->
<ImageLoad onmatch="exclude"/>
<!-- Event ID 8 == CreateRemoteThread. -->
<CreateRemoteThread onmatch="include">
<TargetImage condition="image">lsass.exe</TargetImage>
</CreateRemoteThread>
<!-- Event ID 9 == RawAccessRead. -->
<RawAccessRead onmatch="include"/>
<!-- Event ID 10 == ProcessAccess. -->
<ProcessAccess onmatch="include">
<TargetImage condition="image">lsass.exe</TargetImage>
</ProcessAccess>
<!-- Event ID 11 == FileCreate. -->
<FileCreate onmatch="include"/>
<!-- Event ID 12,13,14 == Regobject added/deleted, Regvalue Set, Regobject Renamed. -->
<RegistryEvent onmatch="include"/>
<!-- Event ID 15 == FileStream Created. -->
<FileStreamHash onmatch="include"/>
<!-- Event ID 17 == PipeEvent. -->
<PipeEvent onmatch="include"/>
</EventFiltering>
</Sysmon>
```

Here we added the onmatchc="exclude" for the NetworkConnection and the ImageLoad events so all of events are logged that has Network connection and loading images into memory.

Save it and lets load it

```
c:\Tools\sysmon>Sysmon.exe -c mimi.xml

System Monitor v6.10 - System activity monitor
Copyright (C) 2014-2017 Mark Russinovich and Thomas Garnier
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 3.40
Configuration file validated.
Configuration updated.
```

Its loaded, now lets check it to make sure it's the way we want it:

```
current configuration:
- Service name: Sysmon
- Driver name: SysmonDrv
- HashingAlgorithms: SHA1,MD5,SHA256,IMPHASH
- Network connection: enabled
- Image loading: enabled
- CRL checking: disabled
- Process Access: enabled

Rule configuration (version 3.40):
- ProcessCreate onmatch: include
- FileCreateTime onmatch: include
- NetworkConnect onmatch: exclude
- ProcessTerminate onmatch: include
- DriverLoad onmatch: include
- ImageLoad onmatch: exclude
- CreateRemoteThread onmatch: include
    TargetImage filter: image value: 'lsass.exe'
- RawAccessRead onmatch: include
- ProcessAccess onmatch: include
    TargetImage filter: image value: 'lsass.exe'
- FileCreate onmatch: include
- RegistryEvent onmatch: include
- RegistryEvent onmatch: include
- RegistryEvent onmatch: include
- FileCreateStreamHash onmatch: include
- PipeEvent onmatch: include
- PipeEvent onmatch: include
```

It looks the way we want it, but we also see the “CreateRemoteTThread” which will not help us at all in detecting mimkatz.exe as its not injecting it self to the lsass.exe, its accessing it.

Lets run the netkatz.exe:

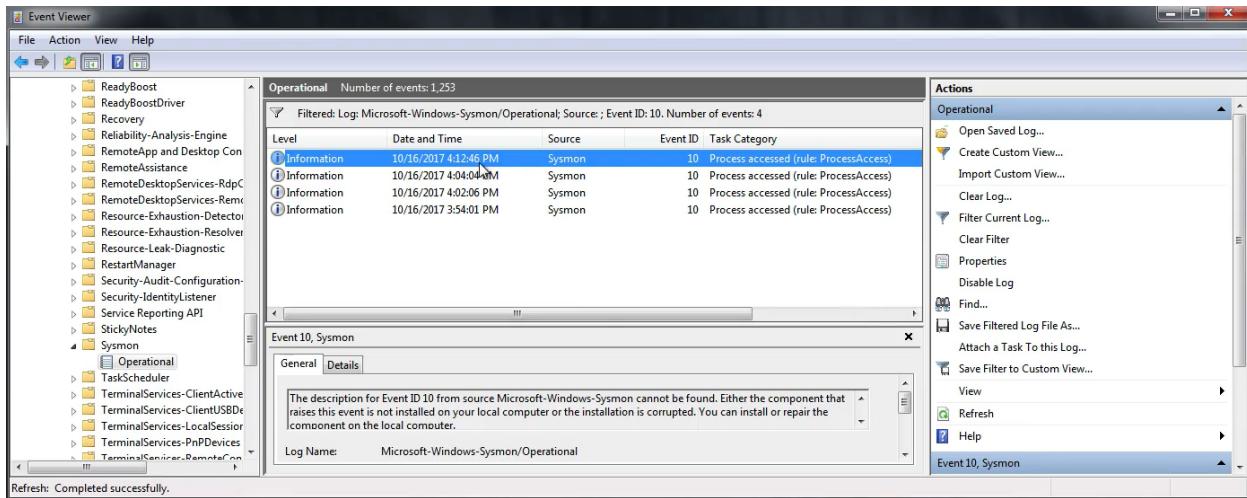
```
c:\Tools\Mimikatz\Netkatz>netkatz.exe privilege::debug sekurlsa::logonpasswords
```

```
Loaded ADVAPI32.dll
Loaded CRYPT32.dll
Loaded cryptdll.dll
Loaded NETAPI32.dll
Loaded ole32.dll
Loaded OLEAUT32.dll
Loaded RPCRT4.dll
Loaded SHLWAPI.dll
Loaded SAMLIB.dll
Loaded Secur32.dll
Loaded SHELL32.dll
Loaded USER32.dll
Loaded USERENV.dll
Loaded VERSION.dll
Loaded HID.DLL
Loaded SETUPAPI.dll
Loaded WinsCard.dll
Loaded WINSTA.dll
Loaded WLDAP32.dll
Loaded advapi32.dll
Loaded msasn1.dll
Loaded ntdll.dll
Loaded netapi32.dll
Loaded KERNEL32.dll
Loaded msrvct.dll
Executing Mimikatz
```

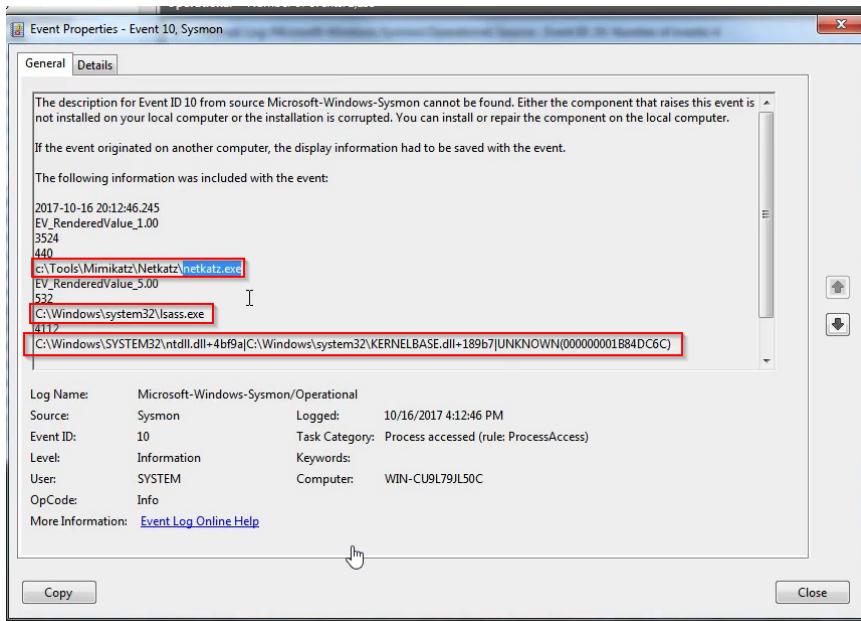
In the output of the netkatz.exe tool we here we see all the loaded DLLs by the Netkatz.exe into the memory:

Now lets check the Event Viewer for the logs:

Here we will filter for Event ID 10 to see the events related to a process access another process:



We see 4 events (idk why, maybe error in the demo), lets double click the first one and see its details:



Here we see the netkatz.exe accessed the lsass.exe process, but the dll part is deferent.

We can look at this dll part in the powershell.exe

```
PS C:\windows\system32> Get-WinEvent -FilterHashtable @{"logname="Microsoft-Windows-Sysmon/Operational";id=10}
| %{$_.Properties[9].Value}
C:\Windows\SYSTEM32\ntdll.dll+4bf9a|C:\Windows\system32\KERNELBASE.dll+189b7|UNKNOWN(000000001B84DC6C)
C:\Windows\SYSTEM32\ntdll.dll+4bf9a|C:\Windows\system32\KERNELBASE.dll+189b7|UNKNOWN(000000001B00DC6C)
C:\Windows\SYSTEM32\ntdll.dll+4bf9a|C:\Windows\system32\KERNELBASE.dll+189b7|UNKNOWN(000000001B13DC6C)
C:\Windows\SYSTEM32\ntdll.dll+4bf9a|C:\Windows\system32\KERNELBASE.dll+189b7|C:\Tools\Mimikatz\x64\mimikatz.exe+6dc6c|c:\Tools\Mimikatz\x64\mimikatz.exe+6bf9a|C:\Tools\Mimikatz\x64\mimikatz.exe+6db91|c:\Tools\Mimikatz\x64\mimikatz.exe+4ae04|c:\Tools\Mimikatz\x64\mimikatz.exe+4ac3a|c:\Tools\Mimikatz\x64\mimikatz.exe+4a98f|c:\Tools\Mimikatz\x64\mimikatz.exe+73935|C:\Windows\system32\kernel32.dll+159cd|C:\Windows\SYSTEM32\ntdll.dll+2a561
PS C:\windows\system32>
```

The first 3 parts looks like belong to 3 events of Netkatz.exe, but lets confirm this with getting the index 4 which is shows the process that did the accessing:

```

PS C:\Windows\system32> Get-WinEvent -FilterHashtable @{"logname="Microsoft-Windows-Sysmon/Operational";id=10}
| %{$_.Properties[4].Value}
c:\Tools\mimikatz\Netkatz\Netkatz.exe
c:\Tools\mimikatz\Netkatz\Netkatz.exe
c:\Tools\mimikatz\Netkatz\Netkatz.exe
c:\Tools\mimikatz\x64\mimikatz.exe
PS C:\Windows\system32>

```

And we are right, from that 4 events, 3 of them are netkatz.exe (maybe the demo had problems so the netkatz.exe was tried 2 times and then it was recorded on the 3rd time.)

And we saw that netkatz.exe loaded images(files) into memory, and its event ID is 7:

The screenshot shows the Windows Event Viewer interface. On the left, a list of events is displayed in a grid format. The columns are Level, Date and Time, Source, Event ID, and Task Category. Most events have Level as Information, Date and Time as 10/16/2017 4:13:16 PM, Source as Sysmon, Event ID as 7, and Task Category as Image loaded (rule: ImageLoad). A red box highlights the Event ID and Task Category column. On the right, a context menu titled 'Operational' is open, listing options like Open Saved, Create Cust, Import Cust, Clear Log..., Filter Current, Properties, Disable Log, Find..., Save All Eve, and Attach a Ta. Below the main grid, a specific event is selected, showing its properties in a detailed view. The 'Details' tab is selected, displaying the following information:

The description for Event ID 7 from source Microsoft-Windows-Sysmon cannot be found. Either the component that raises this event is not installed on your local computer or the installation is corrupted. You can install or repair the component on the local computer.

If the event originated on another computer, the display information had to be saved with the event.

The following information was included with the event:

```

2017-10-16 20:13:16.056
EV_RenderedValue_1.00
2472
C:\Windows\System32\taskhost.exe
C:\Windows\System32\api-ms-win-downlevel-user32-l1-1-0.dll
SHA1=04654DC51565256164AB3BB19B0D7506D7F1B1CC,MD5=72723D3E4781BADC62C3180C137E7B23,SHA256=
0BDA5292928578C5DA79C761E1588A892B9D4A3DA26D3635E714797C653CF492,IMPHASH=00000000000000000000000000000000
true
Microsoft Windows
Valid

```

Below this, a summary table provides the following details:

Log Name:	Microsoft-Windows-Sysmon/Operational		
Source:	Sysmon	Logged:	10/16/2017 4:13:16 PM
Event ID:	7	Task Category:	Image loaded (rule: ImageLoad)
Level:	Information	Keywords:	
User:	SYSTEM	Computer:	WIN-CU9L79JL50C
OpCode:	Info		

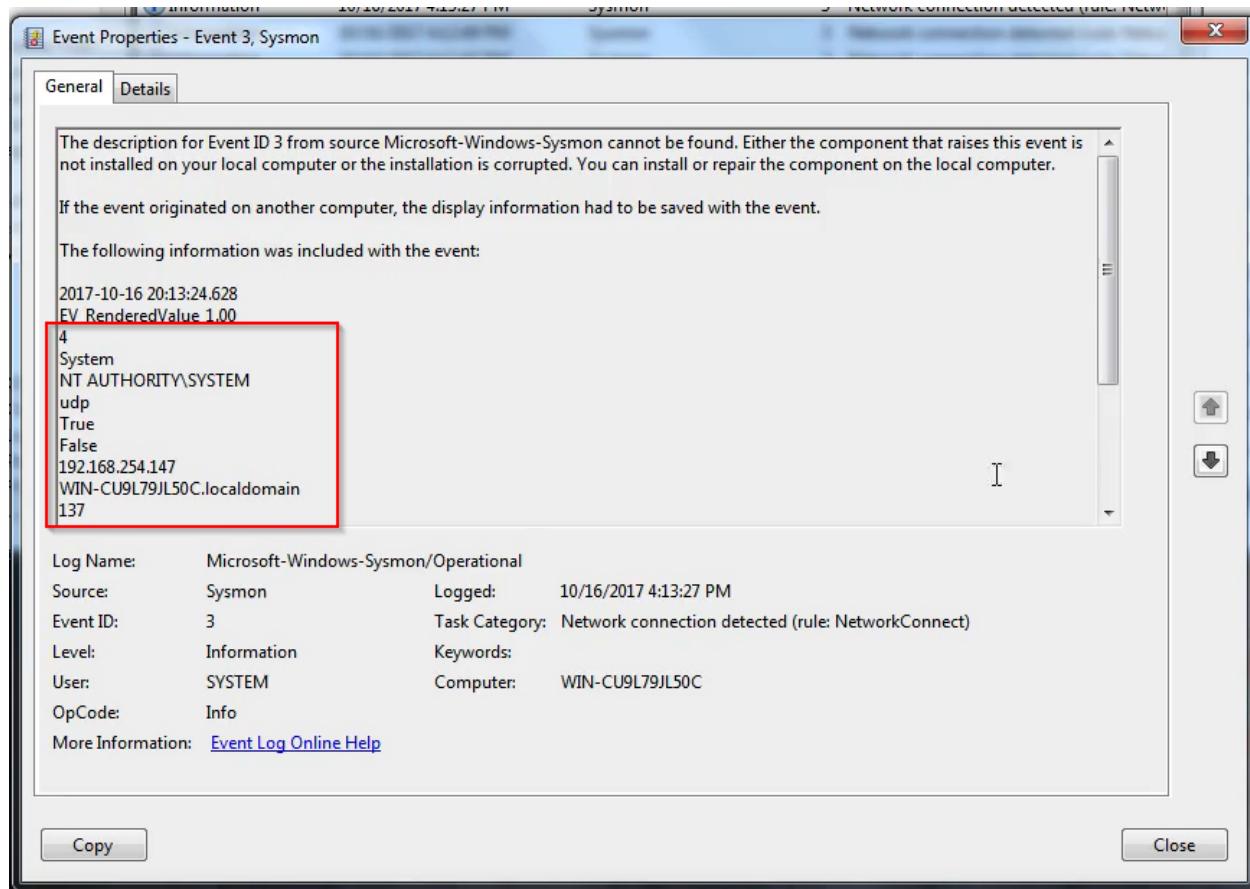
More Information: [Event Log Online Help](#)

On the far right of the event details window, there are two red-bordered buttons: an upward-pointing arrow and a downward-pointing arrow, used for navigating between events.

Here we see the Event ID 7 events and to go through the events UP and Down we can click those 2 buttons. And here we see that that the process(taskhost.exe) that loaded this image "api-ms-win-downl-level-user32-l1-1-0.dll"

Now lets filter for event ID 3 to see the network connections IDs which we know the Netkatz.exe will go the internet and grab the mimkatz and run it in memory.

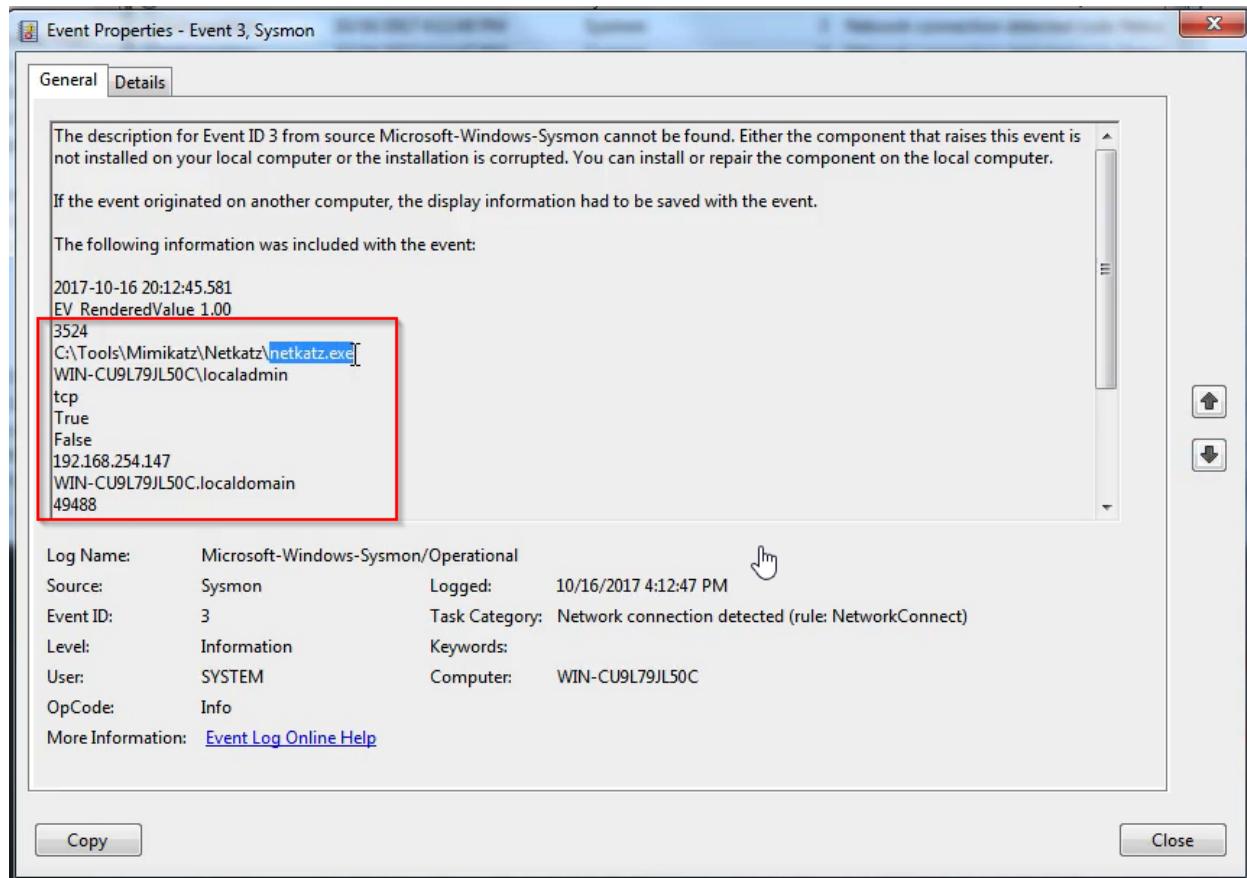
Here we will look at an event just to get an idea of how the logs of Network connection are in the sysmon:



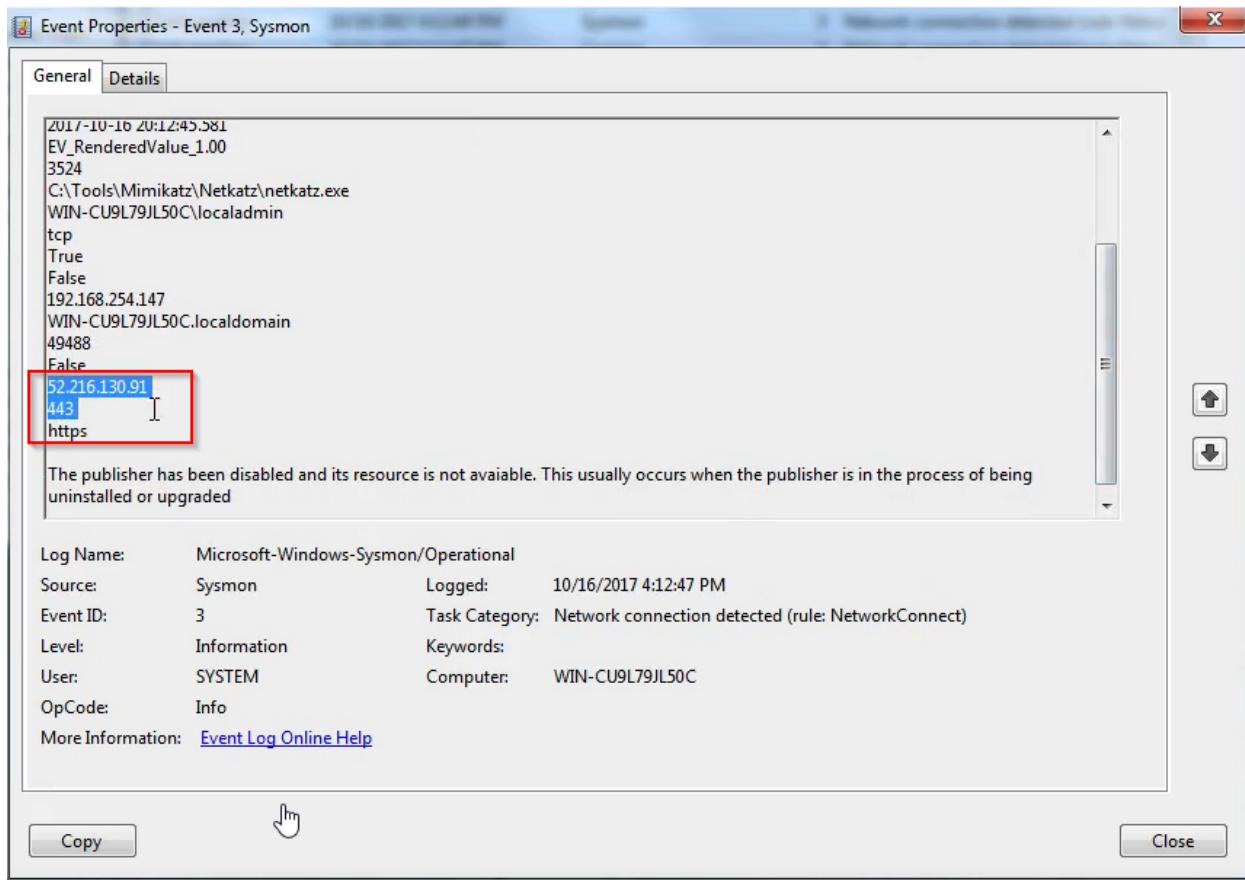
Here first we see the PID of the process that made the connection, then the process name, and then the User, and then the Protocol, and then we see the Host IP (this computer IP mean the IP of the host that made the connection), and at the bottom we see the port used to make this connection which is 137.

If scroll down we will see the details of the host IP that is receiving the connection and the port and protocol like HTTPS, FTP.

Now lets look at the log where the connection is made by netkatz.exe:



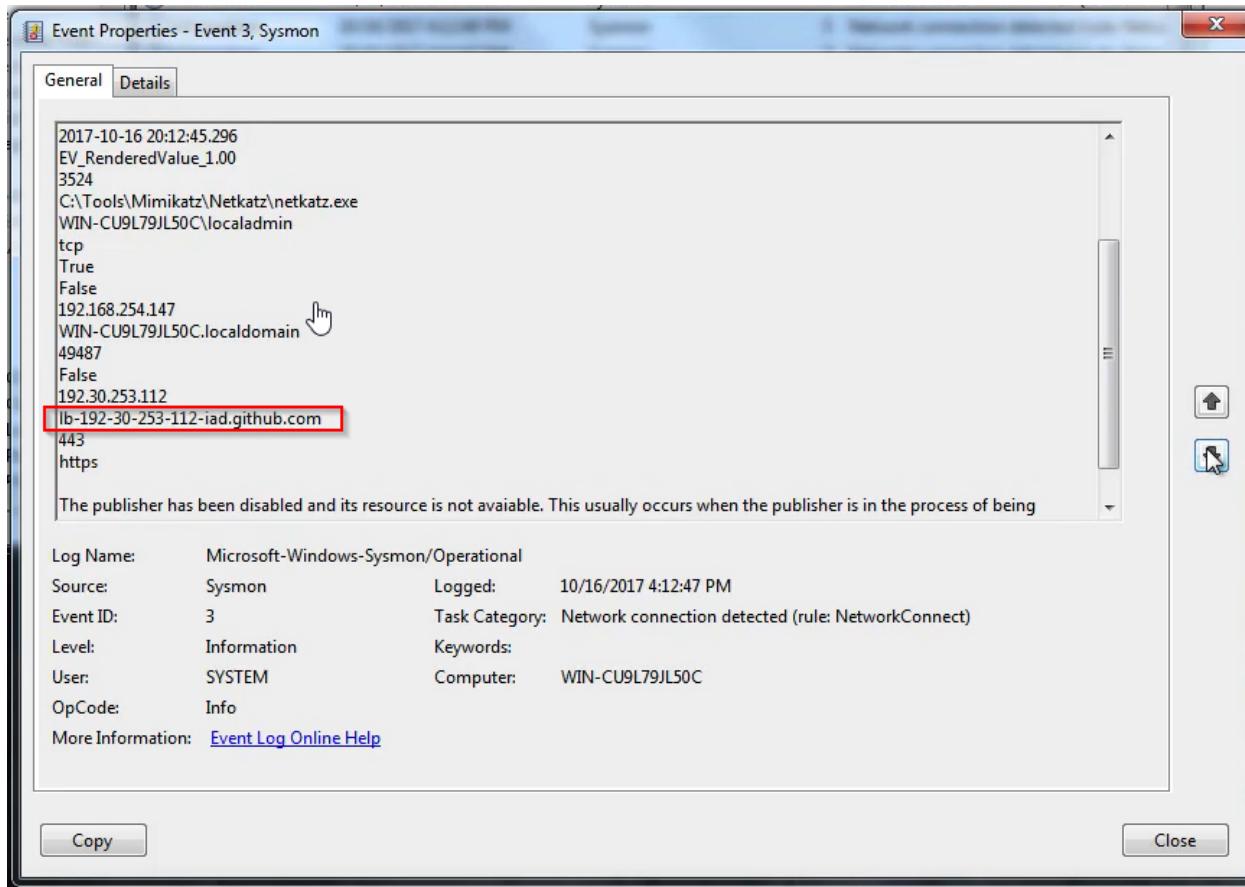
Here we see the process that made this connection has PID 3524 and its name is netkatz.exe, we see it used protocol TCP and the then we see the IP of the host that run the netkatz.exe, and it started the connection on port 49488 and lets scroll down this window to see more details:



Here we see the remote host, or the server that netkaze.exe made the request to, and it used to port 443 and its HTTPS protocol.

So the port 49488 was open by client to communicate , mean the computer that run the netkaze.exe and to the address it communicated to had port 443 open over HTTPS.

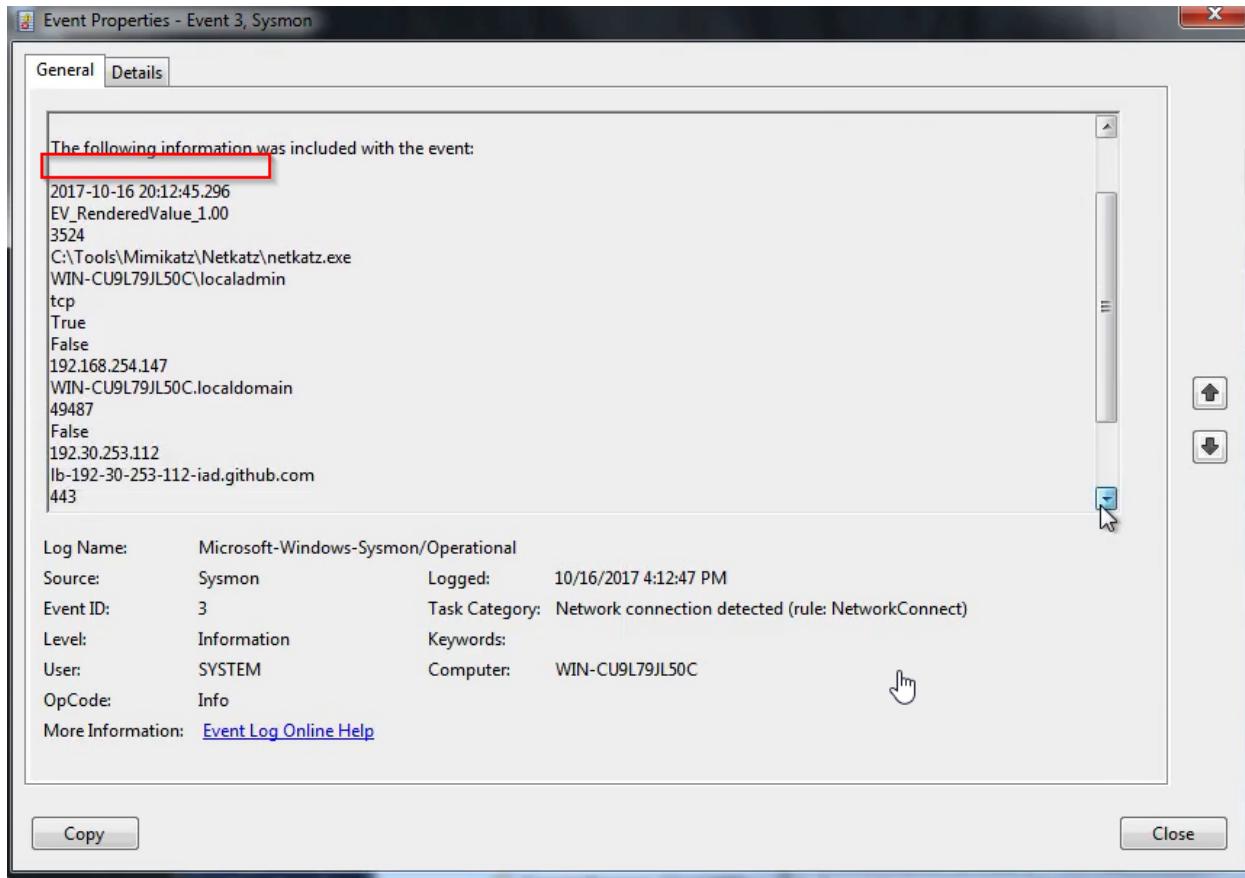
After looking we see this:



We see it went to the github.com, maybe to get the mimkatz from the github.com and other tools from other address and then loaded into the memory. So that's why we see the other address in the screen show about this that has IP 52.*.

And if we count here, we will see the IP address of the server this host communicated is in line 13 which is index 12? No there is an empty line as well , so to look for this in the powershell, we will have to put 13 in the properties.

The empty line:



NOTE: we can always check, if we didn't get the value we wanted, we will try one up or down index:

Lets try 13:

```
PS C:\Windows\system32> Get-WinEvent -FilterHashtable @{Logname="Microsoft-Windows-Sysmon/operational";id=3}
%{$_.Properties[13].Value}
192.168.254.2
52.216.130.91
52.216.130.91
192.30.253.112
192.168.254.2
52.216.226.24
192.30.253.113
192.168.254.2
192.168.254.2
fe80::0:0:4c4f:18b9:e7cd:6d75
PS C:\Windows\system32>
```

Here we got all the IPs this host communicated with, if this was real scenario then we would blacklist those IPs.

To get the Process that made those connection we would put the index location 3,

```
Get-EventLog -LogName "Microsoft-Windows-Sysmon/Operational" | Where-Object { $_.Properties[3].Value -eq "C:\Windows\System32\svchost.exe" } | Select-Object -Property TimeCreated, ProcessId, TaskCategory, Message | Format-Table -AutoSize
```

PS C:\Windows\system32> Get-WinEvent -FilterHashtable @{"logname="Microsoft-Windows-Sysmon/Operational";id=3"} | %{\$_.Properties[3].Value}
C:\Windows\System32\svchost.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\svchost.exe
C:\Windows\System32\svchost.exe
System
System
C:\Tools\Mimikatz\Netkatz\Netkatz.exe
C:\Tools\Mimikatz\Netkatz\Netkatz.exe
C:\Windows\System32\svchost.exe
C:\Tools\Mimikatz\Netkatz\Netkatz.exe
C:\Tools\Mimikatz\Netkatz\Netkatz.exe
C:\Windows\System32\svchost.exe
System
C:\Windows\System32\svchost.exe
PS C:\Windows\system32>

We see that 4 of network connections are made by the netkatz.exe